Hindawi
Security and Communication Networks
Volume 2021, Article ID 6537678, 13 pages
https://doi.org/10.1155/2021/6537678

Research Article

Improved Secure and Lightweight Authentication Scheme for Next-Generation IoT Infrastructure

Chien-Ming Chen and Shuangshuang Liu

College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, China

Correspondence should be addressed to Shuangshuang Liu; shuangliu0309@163.com

Received 25 June 2021; Revised 1 September 2021; Accepted 14 September 2021; Published 30 September 2021

Academic Editor: Azeem Irshad

The Internet of Things (IoT) is a huge network formed by connecting various information sensing devices through the Internet. Although IoT has been popularized in many fields, connected devices can be used only when network security is guaranteed. Recently, Rana et al. proposed a secure and lightweight authentication protocol for the next-generation IoT infrastructure. They claim that their protocol can resist major security attacks. However, in this study, we prove that their protocol is still vulnerable to offline password guessing attacks and privilege internal attacks. In order to solve these shortcomings, we propose an improved protocol, which is proved to be secure by formal and informal analysis. In addition, after comparing the time and memory consumption with other protocols, we find that our protocol has more advantages.

1. Introduction

In recent years, the Internet of Things (IoT) [1–4] has become popular in our everyday life. IoT refers to the real-time collection of any information that needs to be monitored, connected, and interacted with through the use of various devices and technologies such as sensors, radio frequency identification technology, global positioning system, and laser scanners. In the IoT environment, every object (virtual or physical) can be perceived, identified, accessed, and interconnected in a dynamic, ubiquitous network through the Internet. IoT brings great convenience to our lives. Vehicular ad hoc networks [5, 6] are considered to be one of the most promising applications of IoT. They allow people, vehicles, and roadside units to cooperate closely. IoT is also applied to medical healthcare, which is also closely related to our lives. Through the use of IoT, medical healthcare environments have taken on a new look. In an IoT-enabled healthcare system [7–9], wearable sensors can be used to collect information about patients and the surrounding environment. Another example of an IoT application is the smart home [10, 11]. Smart homes improve people's lifestyles and make them more comfortable, safer, and more efficient. In addition, the cloud system based on IoT can help the national government manage some resources to a great extent. The management data through the cloud system greatly reduces human resources and greatly improves the utilization rate of resources. These advantages are mainly based on the principle of the cloud-based Internet of Things. The application of such technology supports legitimate users to access normal data from hospitals, homes, borders, and other areas, which can better manage data to a certain extent.

Because IoT has grown so seamlessly, many end users are ignorant of the existence of these devices. Due to their invisibility, IoT device security is crucial, yet challenging to manage. Several IoT networks have recently been taken over to carry out malicious attacks. For these reasons, addressing these IoT security challenges is critical to their successful development. However, there has been a significant expansion in the number of IoT devices. Designing security mechanisms for all of these devices is complicated due to the heterogeneity and complexity of IoT networks.

For an IoT network to be secure, all the entities (servers, end users, and devices) must mutually authenticate their identities. In addition, all communication should be encrypted to maintain data confidentiality. This means that

a common session key for both sides of the communication is required. Therefore, designing a secure and efficient authenticated key agreement (AKA) scheme is crucial [12–15].

Various AKA schemes for IoT have been proposed. In 2004, Kumari et al. [16] found that Chang et al.'s scheme [17] is vulnerable to offline password-guessing attacks, internal attacks, and server masquerading attacks. They also pointed out that the protocol [17] has security vulnerabilities during the password update phase. To overcome these security weaknesses, Kumari et al. designed an improved scheme. Kumari et al. claimed that their scheme is more secure, more efficient, and more suitable for real-life IoT network use. However, Kaul and Awasthi [18] discovered that Kumari et al.'s protocol [16] is still vulnerable to some attacks. In their scheme, attackers can easily capture some security parameters transmitted on a public channel and then calculate the session key. In response to this, Kaul and Awasthi [18] proposed a robust and secure user authentication protocol based on resource-friendly symmetric cryptographic primitives. Unfortunately, Rana et al. [19] proved that the protocol [18] cannot resist various types of attacks. Therefore, they proposed a secure, lightweight AKA scheme for next-generation IoT infrastructure.

In this study, however, we found that Rana et al.'s scheme [19] is still vulnerable to offline password-guessing attacks and privileged-insider attacks. In their scheme [19], an illegal insider or malicious attacker can calculate the session key or guess passwords if they can capture a user's smart card. Therefore, we propose a new AKA scheme. In the proposed scheme, we utilize the biological information of the users because it is difficult for attackers to obtain this information. To demonstrate that the proposed scheme is indeed secure, we analyze it using Burrows–Abadi–Needham (BAN) logic [20] and also show that it is secure against various types of attacks. Compared with the previous scheme, the proposed scheme has better performance in terms of memory overhead.

The remainder of this paper is organized as follows. In Section 2, we briefly review the scheme proposed by Rana et al. [19]. Section 3 demonstrates that Rana et al.'s scheme [19] is vulnerable to offline password-guessing attacks and privileged-insider attacks. Our proposed scheme is described in Section 4. Sections 5 and 6 provide security and performance analyses and comparisons. Finally, Section 7 concludes the paper.

2. Review of Rana et al.'s Scheme

In this section, we briefly review Rana et al.'s AKA scheme. Their scheme contains three phases: user registration, login, and authentication, and the steps of their scheme are described below. Notations used in this paper are listed in Table 1.

2.1. User Registration Phase.

(1) First, the user U_c selects their own identification ID_c, password PW_c, and an arbitrary number b. Then, the following is calculated:

$$RPW_c = h(m \parallel PW_c), \tag{1}$$

and $\{ID_c, RPW_c\}$ is transmitted to the server through a secure channel.

(2) After the server receives the information from the user, it selects an arbitrary number y_c and calculates

$$\begin{aligned} DI\ D_c &= Enc_{ds}(ID_c \parallel y_c),\\ \alpha_c &= h(ID_c \oplus a) \parallel b,\\ \beta_c &= \alpha_c \oplus h(ID_c \oplus RPW_c),\\ \gamma_c &= y_c \oplus h(\alpha_c \oplus RPW_c),\\ \chi_c &= h(ID_c \parallel RPW_c \parallel y_c \parallel \alpha_c). \end{aligned} \tag{2}$$

(3) Then, the server stores the parameters $\{\beta_c, \gamma_c, \chi_c, DI\ D_c, h(\cdot)\}$ in the smart card memory and sends them to the user U_c through a secure channel.

(4) Finally, the user calculates

$$\eta_c = h(ID_c \oplus PW_c) \parallel m, \tag{3}$$

and stores η_c in the smart card. Now, the smart card contains parameters $\{\beta_c, \gamma_c, \chi_c, \eta_c, DI\ D_c, h(\cdot)\}$.

2.2. Login Phase. When a registered user wants to log in to the system, they perform the following operations:

(1) User U_c enters their ID_c' and PW_c' and inserts the smart card

(2) The smart card reader extracts parameters $m = \eta_c \oplus h(ID_c' \oplus PW_c')$ and $RPW_c' = h(m \parallel PW_c')$

(3) Further, the smart card reader can extract parameters $\alpha_c' = \beta_c \oplus h(ID_c' \oplus RPW_c')$ and $y_c' = \gamma_c \oplus h(\alpha_c' \oplus RPW_c')$ and calculate

$$\chi_c' = h(ID_c' \parallel RPW_c' \parallel y_c' \parallel \alpha_c'). \tag{4}$$

If $\chi_c' = \chi_c$, it means that the legitimate user is allowed to log in; otherwise, the login is refused

(4) After verifying the legitimacy of the user, the reader calculates

$$\begin{aligned} \omega_c &= y_c \oplus (ID_c' \oplus \alpha_c') \oplus h(ID_c' \oplus \alpha_c' \oplus T_1),\\ \nu_c &= h(ID_c' \parallel \alpha_c' \parallel y_c \parallel (\alpha_c' \oplus y_c) \parallel T_1). \end{aligned} \tag{5}$$

The reader then sends the login request $\{DI\ D_c, \omega_c, \nu_c, T_1\}$ to the server through a secure channel.

2.3. Authentication Phase. In this phase, the smart card reader and server authenticate each other by performing the following steps:

(1) S first verifies the validity of the timestamp by calculating T_2-T_1. If the calculated value is less than the given threshold δT, the login request proceeds; otherwise, it is rejected.

Table 1: Notations used in the proposed scheme.

Notations	Descriptions
U_c	c_{th} legal user
ID_c	c_{th} user identity
S	Legal server
PW_c	c_{th} user password
a, b	Private key and number of server
y_c	Arbitrary number for U_c
SC_c	User's smart card
T_1	Time stamp obtained at user's side
T_2	Server's current time stamp
T'	Threshold value
δT_c	Time of transmission delay
$\oplus$	XOR operator
$\|$	Concatenation operator
$h(\cdot)$	Noncollision hash function
SK	Session key
$\mathscr{A}$	The attacker
R_i	Biometric of U_c
ds	Long-term key
$\Rightarrow$	Private communication channel
$\longrightarrow$	Public communication channel

(2) After that, S extracts and calculates ID_c' using $(ID_c' \| y_c) = \mathrm{Dec}_{ds}(DI\ D_c)$ and then calculates the values:

$$\begin{aligned} \alpha_c' &= h(ID_c' \oplus a) \| b, \\ y_c' &= \omega_c' \oplus (ID_c' \oplus \alpha_c') \oplus h(ID_c' \oplus \alpha_c' \oplus T_1), \\ \nu_c' &= h(ID_c' \| \alpha_c' \| y_c' \| (\alpha_c' \oplus y_c') \| T_1). \end{aligned} \tag{6}$$

Then, S verifies the validity of the login by comparing the calculated ν_c' with the stored ν_c. If the two are equal, the verification passes; otherwise, the verification fails and the server refuses to accept the login request.

(3) After verifying the correctness of ν_c, the server continues to calculatez

$$\mu_c = h(ID_c' \| y_c' \| (\alpha_c' \oplus y_c') \| T_2). \tag{7}$$

Then, S sends the calculated μ_c and timestamp T_2 to U.

(4) When U receives the information from the server, it first verifies the validity of T_2 and then calculates

$$\mu_c' = h(ID_c \| y_c \| (\alpha_c \oplus y_c) \| T_2). \tag{8}$$

U checks whether μ_c' is equal to μ_c. If so, S is successfully verified.

(5) Finally, after mutual verification, the session key SK can be calculated:

$$SK = h(ID_c \oplus \alpha_c \oplus y_c \oplus T_1 \oplus T_2). \tag{9}$$

3. Cryptanalysis of Rana et al.'s Scheme

In this section, we first describe the threat model. Then, we show that Rana et al.'s scheme is insecure against offline password-guessing attacks and privileged-insider attacks.

3.1. Threat Model. This threat pattern shows the capabilities of an adversary, which are also considered and discussed in [21, 22]. $\mathscr{A}$'s capabilities are as follows:

(1) $\mathscr{A}$ can perform complete access control on the transmission channel. It can block, change, remove, replay, and hinder the messages passed between participants through a public channel.

(2) $\mathscr{A}$ can get the information stored in the smart card using power analysis [23, 24].

(3) $\mathscr{A}$ can obtain the information in the smart card and the information transmitted by the user on the secure channel during the registration process [25].

(4) $\mathscr{A}$ can simultaneously obtain the information in the smart card and perform offline password guessing as stated in [26, 27].

(5) $\mathscr{A}$ can know any two of the user's password, smart card, and biological information.

(6) $\mathscr{A}$ can obtain the session key that the user communicated with the server before.

(7) $\mathscr{A}$ can register as a legitimate user in a legitimate way.

3.2. Offline Password-Guessing Attack.

(1) First, the attacker $\mathcal{A}$ steals the smart card and gets the information $\{\beta_c^N, \gamma_c^N, \chi_c^N, \eta_c^N\}$

(2) $\mathcal{A}$ guesses the user's ID_c and PW_c at the same time

(3) According to the user's ID_c, password PW_c, and η_c and γ_c values obtained from the smart card, $\mathcal{A}$ calculates

$$\begin{aligned} m &= \eta_c \oplus h(ID_c \oplus PW_c), \\ RPW_c &= h(m \parallel PW_c), \\ \alpha_c &= \beta_c \oplus h(ID_c \oplus RPW_c), \\ y_c &= \gamma_c \oplus h(\alpha_c \oplus RPW_c). \end{aligned} \tag{10}$$

(4) Finally, $\mathcal{A}$ obtains the session key SK according to the value of α_c and y_c calculated above:

$$SK = h(ID_c \oplus \alpha_c \oplus y_c \oplus T_1 \oplus T_2). \tag{11}$$

3.3. Privileged-Insider Attack.

(1) First, the attacker $\mathcal{A}$ steals the smart card and gets the information $\{\beta_c^N, \gamma_c^N, \chi_c^N, \eta_c^N\}$

(2) Then, privileged insiders can obtain the information ID_C and RPW_c of legitimate users during registration

(3) $\mathcal{A}$ can calculate the following parameters by using the information β_c obtained in the smart card and the information ID_C and RPW_c obtained during user registration:

$$\begin{aligned} \alpha_c &= \beta_c \oplus h(ID_c \oplus RPW_c), \\ y_c &= \gamma_c \oplus h(\alpha_c \oplus RPW_c). \end{aligned} \tag{12}$$

(4) Finally, the attacker can calculate the session key SK according to the above parameters:

$$SK = h(ID_c \oplus \alpha_c \oplus y_c \oplus T_1 \oplus T_2). \tag{13}$$

4. Proposed Scheme

In this section, we describe the specific process of the protocol and the overall architecture diagram. The main body of the protocol includes users and servers. The agreement consists of four phases: user registration, login, authentication, and password change. Figure 1 illustrates the architecture of the proposed protocol. User represents the main participant in the communication, and server represents the entity that communicates with the user.

4.1. User Registration Phase. Figure 2 illustrates the user registration phase. The detailed steps are as follows:

(1) First, U_c selects their ID_c, password PW_c, and bio information R_i, as well as an arbitrary number m, to calculate

$$BRPW_c = (h(Ri) \oplus PW_c) \parallel m. \tag{14}$$

Then, ds is used to encrypt ID_c, with the result:

$$DI\ D_c = \mathrm{Enc}_{ds}(ID_c). \tag{15}$$

U_c then transmits $\{DI\ D_c, BRPW_c\}$ to S through a secure channel.

(2) After receiving the information from U, S selects an arbitrary number y_c to decrypt $DI\ D_c$, obtains the value of ID_c, and then calculates

$$\begin{aligned} ID_c &= De\ c_{ds}(DI\ D_c), \\ \alpha_c &= h(ID_c \oplus a) \parallel b, \\ \beta_c &= \alpha_c \oplus h(ID_c \oplus BRPW_c), \\ \gamma_c &= y_c \oplus h(\alpha_c \oplus BRPW_c), \\ \chi_c &= h(ID_c \parallel BRPW_c \parallel y_c \parallel \alpha_c). \end{aligned} \tag{16}$$

(3) Finally, the calculated parameters $\{\beta_c, \gamma_c, \chi_c, DI\ D_c, h(\cdot)\}$ are stored in the smart card, and S sends the smart card to U through a secure channel. U calculates η_c after receiving the message:

$$\eta_c = R_i \oplus m \oplus h(ID_c \oplus PW_c). \tag{17}$$

Then, η_c is saved in the smart card, and the registration process of the user is complete.

4.2. Login Phase.

(1) U enters their own ID_c', PW_c', and bio information R_i.

(2) After inputting the information, calculate

$$\begin{aligned} m &= \eta_c \oplus R_i \oplus h(ID_c' \oplus PW_c'), \\ BRPW_c' &= (h(R_i) \oplus PW_c') \parallel m, \\ \alpha_c' &= \beta_c \oplus h(ID_c' \oplus BRPW_c'), \\ y_c' &= \gamma_c \oplus h(\alpha_c' \oplus BRPW_c'), \\ \chi_c' &= h(ID_c' \parallel BRPW_c' \parallel y_c' \parallel \alpha_c'). \end{aligned} \tag{18}$$

Then, verify whether χ_c' and χ_c are equal. If they are equal, the verification passes; otherwise, the login request sent by U to S is rejected.

(3) If the verification passes, the reader will calculate

$$\begin{aligned} \omega_c &= y_c' \oplus h(ID_c' \oplus \alpha_c') \oplus h(ID_c' \oplus \alpha_c' \oplus T_1), \\ v_c &= h(ID_c' \parallel \alpha_c' \parallel y_c' \parallel (\alpha_c' \oplus y_c') \parallel T_1). \end{aligned} \tag{19}$$

Then, the login request $\{DI\ D_c, \omega_c, v_c, T_1\}$ is sent to the server.

4.3. Authentication Phase. This section describes the process of mutual authentication between S and U. After the user sends the login request to the server, the server starts to

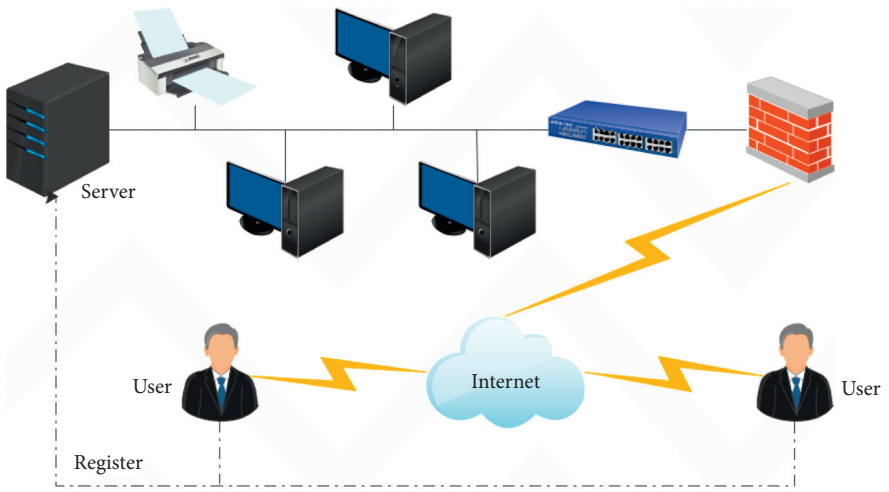

Figure 1: Network architecture.

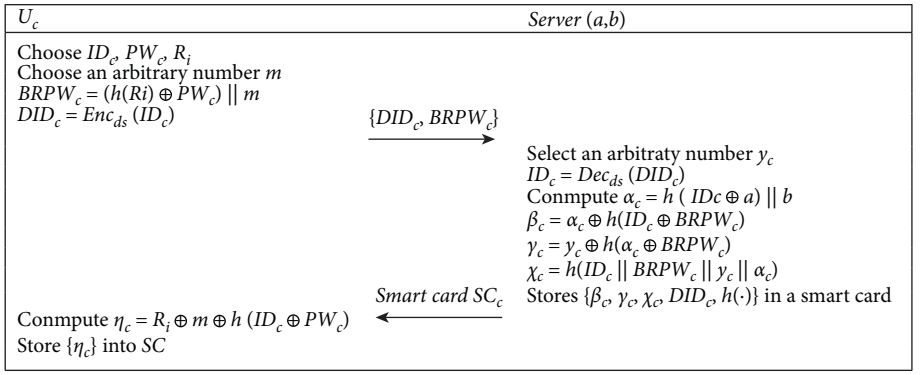

Figure 2: User registration phase.

verify whether U is legitimate by calculating a series of parameters, and U verifies the validity of S by calculating the values of some parameters. The authentication process is described in detail below. The login phase and authentication phase are shown in Figure 3.

(1) After S receives the request from U, it first verifies whether the present timestamp is reasonable. It then decrypts $DI\ D_c$ to obtain ID_c and calculates

$$\begin{aligned} \alpha_c' &= h((ID_c' \oplus a) \| b), \\ y_c' &= \omega_c' \oplus h(ID_c' \oplus \alpha_c') \oplus (ID_c' \oplus \alpha_c' \oplus T_1), \\ v_c' &= h(ID_c' \| \alpha_c' \| y_c' \| (\alpha_c' \oplus y_c') \| T_1). \end{aligned} \tag{20}$$

S verifies whether v_c' and v_c are equal. If not, S rejects the login request from U. If equal, S receives the login request from U and then calculates the session key of both sides:

$$SK = h(ID_c' \oplus \alpha_c' \oplus y_c' \oplus T_1 \oplus T_2). \tag{21}$$

(2) After calculating the session key, S continues to calculate

$$\mu_c = h(ID_c' \| y_c' \| (\alpha_c' \oplus y_c') \| T_2). \tag{22}$$

Then, S passes $\{\mu_c, T_2\}$ to U

(3) After receiving the message from S, the user first verifies the validity of the timestamp T_2 and then calculates

$$\mu_c' = h(ID_c \| y_c' \| (\alpha_c' \oplus y_c') \| T_2). \tag{23}$$

U verifies whether μ_c' is equal to μ_c. If it is equal, U calculates the session key:

$$SK = h(ID_c \oplus \alpha_c' \oplus y_c' \oplus T_1 \oplus T_2). \tag{24}$$

Here, the authentication process for U and S is completed.

4.4. Password Change Phase. If U wants to change their password PW_c to PW_c^N, the following steps are performed:

(1) U first inserts their own smart card and enters their ID_c, current password PW_c, bio information R_i, and new password PW_c^N.

$User U_c$	Server S
Input ID_c', PW_c', R_i	
Compute $m = \eta_c \oplus R_i \oplus h(ID_c' \oplus PW_c')$	
$BRPW_c = (h(R_i) \oplus PW_c') \mid\mid m$	
$\alpha_c' = \beta_c \oplus h(ID_c' \oplus BRPW_c')$	
$y_c' = \gamma_c \oplus h(\alpha_c' \oplus BRPW_c')$	
$\chi_c' = h(ID_c' \mid\mid BRPW_c' \mid\mid y_c' \mid\mid \alpha_c')$	
Check $\chi_c' \stackrel{?}{=} \chi_c$	
Compute $\omega_c = y_c \oplus (ID_c' + \alpha_c') \oplus h(ID_c' \oplus \alpha_c' \oplus T_1)$	
$v_c = h(IDc' \mid\mid \alpha_c' \mid\mid y_c \mid\mid (\alpha_c' \oplus y_c) \mid\mid T_1)$	
$DID_c, \omega_c, v_c, T_1 \longrightarrow$	
	$T_1' - T_1 < \Delta T$
	$(ID_c) = Dec_{ds}(DID_c)$
	Compute $\alpha_c' = h(ID_c' \oplus a) \mid\mid b$
	$y_c' = \omega_c' \oplus h(ID_c' \oplus \alpha_c') \oplus h(ID_c' \oplus \alpha_c' \oplus T_1)$
	$v_c' = h(ID_c' \mid\mid \alpha_c' \mid\mid y_c' \mid\mid \alpha_c' \oplus y_c') \mid\mid T_1$
	Check $v_c' \stackrel{?}{=} v_c$
	$SK = h(ID_c \oplus \alpha_c \oplus y_c \oplus T_1 \oplus T_2)$
	$\mu_c = h(ID_c' \mid\mid y_c' \mid\mid \alpha_c' \oplus y_c') \mid\mid T_2$
	$\mu_c, T_2 \longleftarrow$
Checks $T_2' - T_2 < \Delta T$	
Compute $\mu_c' = h(ID_c \mid\mid y_c \mid\mid (\alpha_c \oplus y_c) \mid\mid T_2)$	
Check $\mu_c' = \mu_c$	
$SK = h(ID_c \oplus \alpha_c \oplus y_c \oplus T_1 \oplus T_2)$	
Input $ID_c', PW_c', R_i, PW_c^{new}$	
Compute $m = \eta_c \oplus R_i \oplus h(ID_c' \oplus PW_c')$	
$BRPW_c = (h(R_i) \oplus PW_c') \mid\mid m$	
$\alpha_c' = \beta_c \oplus h(ID_c' \oplus BRPW_c')$	
$y_c' = \gamma_c \oplus h(\alpha_c' \oplus BRPW_c')$	
$\chi_c' = h(ID_c' \mid\mid BRPW_c' \mid\mid y_c' \mid\mid \alpha_c')$	
Check $\chi_c' \stackrel{?}{=} \chi_c$	
Compute $BRPW_c^N = h(m \mid\mid PW_c^N)$	
$\beta_c^N = \alpha_c \oplus h(ID_c \oplus BRPW_c^N)$	
$\gamma_c^N = y_c \oplus h(\alpha_c \oplus BRPW_c^N)$	
$\chi_c^N = h(ID_c \mid\mid BRPW_c^N \mid\mid y_c \mid\mid \alpha_c)$	
$\eta_c^N = h(ID_c \oplus PW_c^N) \oplus m \oplus R_i$	
Update $\beta_c^N, \gamma_c^N, \chi_c^N, \eta_c^N$ on the smart cars	

Figure 3: Login and authentication phase.

(2) According to some parameter values in the smart card and their own identity information, the following are calculated:

$$\begin{aligned} m &= \eta_c \oplus R_i \oplus h(ID_c' \oplus PW_c'), \\ BRPW_c' &= (h(R_i) \oplus PW_c') \parallel m, \\ \alpha_c' &= \beta_c \oplus h(ID_c' \oplus BRPW_c'), \\ y_c' &= \gamma_c \oplus h(\alpha_c' \oplus BRPW_c'), \\ \chi_c' &= h(ID_c' \parallel BRPW_c' \parallel y_c' \parallel \alpha_c'). \end{aligned} \tag{25}$$

If the calculated value of χ_c' is equal to the value of χ_c stored in the smart card, the user is considered legitimate and allowed to change the password.

(3) Some parameter values need to be updated in the process of password modification. The specific calculation process is as follows:

$$\begin{aligned} BRPW_c^N &= \left(h(Ri) \oplus PW_c^N\right) \parallel m, \\ \beta_c^N &= \alpha_c \oplus h\left(ID_c \oplus RPW_c^N\right), \\ \gamma_c^N &= y_c \oplus h\left(\alpha_c \oplus RPW_c^N\right), \\ \chi_c^N &= h\left(ID_c \parallel RPW_c^N \parallel y_c \parallel \alpha_c\right), \\ \eta_c^N &= R_i \oplus m \oplus h\left(ID_c \oplus PW_c^N\right). \end{aligned} \tag{26}$$

(4) Finally, the values $\{\beta_c, \gamma_c, \chi_c, \eta_c\}$ stored in the smart card are updated to the modified values $\{\beta_c^N, \gamma_c^N, \chi_c^N, \eta_c^N\}$, and the process of password modification is completed.

5. Security Analysis

5.1. Formal Security Analysis. Burrows–Abadi–Needham (BAN) logic [20] has been used in several studies to prove whether a protocol can be executed securely. This section uses BAN logic to prove the security and reliability of our proposed protocol. This proof verifies that our protocol can successfully establish and share a session key between the user and server. In the following proof, U represents the user and S represents the server. The specific proof rules and process are as follows:

5.1.1. BAN Logic Rules.

(i) Message-meaning rule (R1): $(U \mid \equiv U \overset{K}{\leftrightarrow} S, P \triangleleft \{M\}_K) / (U \mid \equiv S \mid \sim M)$ and $(U \mid \equiv U \overset{N}{\rightleftharpoons} NS, U \triangleleft \langle M \rangle_N) / (U \mid \equiv S \mid \sim M)$

(ii) Nonce-verification rule (R2): $(U \mid \equiv \sharp(M), U \mid \equiv S \mid \sim M) / (U \mid \equiv S \mid \equiv M)$

(iii) Jurisdiction rule (R3): $(U \mid \equiv S \mid \Rightarrow M, U \mid \equiv S \mid \equiv M) / (U \mid \equiv M)$

(iv) Freshness rule (R4): $(U \mid \equiv \sharp(M)) / (U \mid \equiv \sharp(M, N))$

(v) Belief rule (R5): $(U \mid \equiv M, U \mid \equiv N) / (U \mid \equiv (M, N))$

(vi) Session key rule (R6): $(U \mid \equiv \sharp(M), U \mid \equiv S \mid \equiv M) / (U \mid \equiv U \overset{K}{\leftrightarrow} S)$

5.1.2. Goals.

(i) G1: $U \mid \equiv U \overset{SK}{\leftrightarrow} S$

(ii) G2: $S \mid \equiv U \overset{SK}{\leftrightarrow} S$

(iii) G3: $U \mid \equiv S \mid \equiv U \overset{SK}{\leftrightarrow} S$

(iv) G4: $S \mid \equiv U \mid \equiv U \overset{SK}{\leftrightarrow} S$

5.1.3. Idealizing Communication.

(i) M1: $U \longrightarrow S$: $\{DI \quad D_c, \omega_c, v_c, T_1\}$

(ii) M2: $S \longrightarrow U$: $\{\mu_2, T_2\}$

5.1.4. Initial State Assumptions.

(i) A1: $U \mid \equiv U \overset{ds}{\rightleftharpoons} S$

(ii) A2: $S \mid \equiv U \overset{ds}{\rightleftharpoons} S$

(iii) A3: $S \mid \equiv \sharp(ID_c, \alpha_c, y_c,)$

(iv) A4: $S \mid \equiv U \mid \Rightarrow ID_c$

(v) A5: $S \mid \equiv U \overset{ID_c}{\rightleftharpoons} S$

(vi) A6: $U \mid \equiv U \overset{ID_c}{\rightleftharpoons} S$

(vii) A7: $S \mid \equiv U \mid \Rightarrow (\alpha_c, y_c)$

(viii) A8: $S \mid \equiv \sharp(ID_c, \alpha_c, y_c,)$

(ix) A9: $U \mid \equiv S \mid \Rightarrow (\alpha_c, y_c)$

5.1.5. Detailed Steps.

By considering the message M1 and using the seeing rule, we get

S1: $S \triangleleft \{\langle ID_c \rangle_{ds}, \langle \alpha_c, y_c \rangle_{ds}, T_1\}$.

Using S1, we get

S2: $S \triangleleft \{\langle ID_c \rangle_{ds}\}$.

Under the assumption of A2, using S2, R1 can be used to obtain

S3: $S \mid \equiv U \mid \sim (ID_c)$.

With conclusion S3, using A3 and R2, the following can be obtained:

S4: $S \mid \equiv U \mid \equiv (ID_c)$.

Using A4, R3, and conclusion S4, the following can be obtained:

S5: $S \mid \equiv (ID_c)$.

According to conclusion S1, the following can be obtained:

S6: $S \triangleleft \{\langle \alpha_c, y_c \rangle_{ID_c}\}$.

Using A6, R1, and conclusion S6, the following can be obtained:

S7: $S \mid \equiv U \mid \sim (\alpha_c, y_c)$.

Using A3, R2, and conclusion S7, the following can be obtained:

S8: $S \mid \equiv U \mid \equiv (\alpha_c, y_c)$.

Using A7, R3, and conclusion S8, the following can be obtained:

S9: $S \mid \equiv (\alpha_c, y_c)$.

Because $SK = h(ID_c \oplus \alpha_c \oplus y_c \oplus T_1 \oplus T_2)$, using S5 and S9, we obtain

S10: $S \mid \equiv U \overset{SK}{\leftrightarrow} S$ (G2).

Using A3 and R4, we can obtain

S11: $S \mid \equiv U \mid \equiv U \overset{SK}{\leftrightarrow} S$ (G4).

In addition, considering the message M2, we obtain

S12: $U \lhd \{\langle \alpha_c, y_c \rangle_{ID_c}, T_2\}$.

By using A6, S1, and R1, we obtain

S13: $U \mid \equiv S \mid \sim (\alpha_c, y_c)$.

With conclusion S13, using A8 and applying R2, we obtain

S14: $U \mid \equiv S \mid \equiv (\alpha_c, y_c)$.

Applying A9, S14, and R3, we obtain

S15: $U \mid \equiv (\alpha_c, y_c)$.

Because $SK = h(ID_c \oplus \alpha_c \oplus y_c \oplus T_1 \oplus T_2)$, using S5 and S9, we obtain

S16: $U \mid \equiv S \overset{SK}{\leftrightarrow} S$ (G1).

With conclusion S16, using A8 and R4, we can obtain

S17: $U \mid \equiv S \mid \equiv U \overset{SK}{\leftrightarrow} S$(G3).

5.2. ROR Formal Security Proof

5.2.1. ROR Model. This paper follows the ROR (Random Oracles) model under the proof of security, and two participants U and S are mentioned in the paper. First, let H_U^x and H_S^y as the xth user and yth server, respectively. Then, let $\mathcal{U} = \{H_U^x, H_S^y\}$ and $\mathcal{A}$ can perform the following operations.

Execute$(\mathcal{U})$: by executing this query, $\mathcal{A}$ can get the messages transmitted by U and S through the common channel.

Send$(\mathcal{U}, \mathcal{M})$: with the help of send query, $\mathcal{A}$ can send messages to U and S. In addition, $\mathcal{A}$ can also receive response messages from two participants.

Corrupt$(\mathcal{U})$: with the help of this query, $\mathcal{A}$ can obtain the parameters information stored in the smart card as well as some temporary parameters information and long-term key.

Hash(String): by performing this operation, $\mathcal{A}$ can obtain the value in the hash.

Test$(\mathcal{U})$: this operation is mainly used to verify whether the session key between the user and the server is secure. By tossing a homogeneous coin $\mathcal{C}$, the result of the coin is known only to $\mathcal{A}$. If $\mathcal{C} = 1$, $\mathcal{A}$ can know the correct session key. If $\mathcal{C} = 0$, a null value is an output.

Definition 1 (one-way anticollision hash function): this is a common mathematical function that inputs a variable length field and then produces a fixed length output. If $\mathrm{Adv}(m) = \Pr[(m, n)\varepsilon_R A; h(m) = h(n)] \le t$ for at most run time m, the hash function is considered hash collision proof.

Definition 2 Symmetric encryption method is used in the proposed protocol. Suppose $E_{K_1}, E_{k_2}, \ldots, E_{K_n}$ are encryption methods based on different keys K. In the model, the probability that $\mathcal{A}$ can crack the correct session key is $\mathrm{Adv}_A^K(\eta) = |2\Pr[A \longleftarrow E_{K_1}; (b_0, b_1) \longleftarrow A; \alpha \longleftarrow 0, 1; \beta \longleftarrow E_{K_1}(b_\alpha) : \mathcal{A}(\beta) = \alpha] - 1|$.

Theorem 1. *If $\mathcal{A}$ is a polynomial time η opponent executing our scheme under the ROR model and we choose to look at Zipf's law [28] for the user's password, the possibility of $\mathcal{A}$ damaging the session key is $Adv_A^P(\eta) \le (t_{send} + t_{exe})^2/2^{u-1} + 2\,Adv_A^K(\eta) + t_{hash}^2 \cdot 2^{l-1} + 2\max\{D' \cdot t_{send}^{X'}, t_{send}/2^l\}$ + where l represents the length of the password.*

5.2.2. Security Proof

Proof. In the proof process, we define six games GM_0 to GM_5 and prove the theorem mentioned above according to the defined six game rules. $\mathrm{Succ}_A^{GMi}(\eta)$ represents the probability of $\mathcal{A}'$s success in the game. The specific proof is as follows.

GM_0: in the initial game, $\mathcal{A}$ does not perform any query operations. According to the definition of security primitives, we can get $\mathrm{Adv}_A^P(\eta) = |2\Pr[\mathrm{Succ}_A^{GM0}(\eta)]|$.

GM_1: GM_1 adds the *execute* operation on the basis of GM_0, that is, $\mathcal{A}$ can intercept and tamper with the information transmitted on the public channel $M_1 = \{DI\ D_c, \omega_c, v_c, T_1\}$ and $M_2 = \{\mu_c, T_2\}$. However, $\mathcal{A}$ cannot obtain the session keys of both parties according to the information obtained on the public channel, so the probability of GM_1 is equal to that of GM_0, $\Pr[\mathrm{Succ}_A^{GM1}(\eta)] = \Pr[\mathrm{Succ}_A^{GM0}(\eta)]$.

GM_2: GM_2 adds Hash and Send query operations on the basis of GM_1. According to the birthday paradox, it can be concluded that the maximum probability of hash collision is $t_{\mathrm{hash}}^2/2^{l+1}$. Therefore, it can be concluded that the maximum probability of hash collision of text transmitted by both sides of the session is $(t_{\mathrm{send}} + t_{\mathrm{exe}})^2/2^u$. Finally, we can draw a conclusion $|\Pr[\mathrm{Succ}_A^{GM2}(\eta)] - \Pr[\mathrm{Succ}_A^{GM1}(\eta)]| \le t_{\mathrm{hash}}^2/2^{l+1} + (t_{\mathrm{send}} + t_{\mathrm{exe}})^2/2^u$. The symbol l appearing in the formula represents the length of the hash value and u represents the length of the transmitted text.

GM_3: on the basis of the above game rules, we added the provision that $\mathcal{A}$ can obtain the parameters information stored in the smart card in the new round of game, that is, $\mathcal{A}$ can obtain the parameters $\{\beta_c, y_c, \gamma_c, DI\ D_c\}$ by executing the *Corrupt* operation. On this basis, we perform an offline password guessing attack. First, $\mathcal{A}$ calculates $\alpha_c = \beta_c \oplus h(ID_c' \oplus BRPW')$, $BRPW' = (h(R_i) \oplus PW_c') \parallel m$, but U's identity ID_c and

U's biological information R_i are confidential to us, so they cannot be obtained. According to Zipf's law [28], we can draw a conclusion: $|Pr[Succ_A^{GM3}(\eta)] - Pr[Succ_A^{GM2}(\eta)]| \le max\{D' \cdot t_{send}^{X'}, t_{send}/2^l\}$.

GM_4: in this game rule, we analyze the security of the communication session key between both sides. We mainly analyze it from the following three aspects. The first is to prove that the protocol has perfect forward security. The second is to prove that $\mathscr{A}$ can block the user impersonation attacks. The third is that $\mathscr{A}$ can block the known session-specific temporary information attacks.

Perfect forward security: $\mathscr{A}$ obtains the value of the long-term key ds through $Corrupt$.

Known session-specific temporary information attacks: $\mathscr{A}$ obtains the value of temporary information m or y_c through $Corrupt$ query.

User impersonation attacks: $\mathscr{A}$ obtains the information $\{DI\ D_c, \omega_c.\nu_c, T_1\}$ transmitted by both communication parties through the public channel through Exe query, but U's identity ID_c is obtained by symmetric encryption with the long-term key ds. However, the value of the long-term key ds cannot be obtained.

The session key $SK = h(ID_c \oplus \alpha_c \oplus y_c \oplus T_1 \oplus T_2)$ of both communication parties: in the first case, $\mathscr{A}$ must obtain the values of α_c and y_c in order to obtain the session key, but the value of α_c needs U's biological information. In the second case, $\mathscr{A}$ obtains the value of temporary information, but U's identity ID_c is obtained through symmetric encryption. In the third case, because U's identity ID_c is obtained through symmetric encryption, $\mathscr{A}$ cannot obtain U's real identity, so it is impossible to carry out simulated attacks. Therefore, we can conclude that $|\Pr[\mathrm{Succ}_A^{GM4}(\eta)] - \Pr[\mathrm{Succ}_A^{GM3}(\eta)]| \le A\ DV_A^K(\eta)$.

GM_5: in the final rule of the game, $\mathscr{A}$ uses hash query $h(ID_c \oplus \alpha_c \oplus y_c \oplus T_1 \oplus T_2)$; then, $\mathscr{A}$ can guess the possibility of the session key: $|\Pr[\mathrm{Succ}_A^{GM5}(\eta)] - \Pr[\mathrm{Succ}_A^{GM4}(\eta)]| \le t_{\mathrm{hash}}^2/2^{l+1}$.

As we all know, the probability of guessing the session key correctly is $|\Pr[\mathrm{Succ}_A^{GM5}(\eta)] = 1/2$.

To sum up, we can get it according to the above formula:

$$\begin{aligned}\frac{1}{2}\mathrm{Adv}_A^P(\eta) &= \Pr\left[\mathrm{Succ}_A^{GM_0}(\eta)\right] - \frac{1}{2} = \Pr\left[\mathrm{Succ}_A^{GM_0}(\eta)\right] - \Pr\left[\mathrm{Succ}_A^{GM_5}(\eta)\right] = \Pr\left[\mathrm{Succ}_A^{GM_1}(\eta)\right] \\ &\quad - \Pr\left[\mathrm{Succ}_A^{GM_5}(\eta)\right] \le \Pr\left[\mathrm{Succ}_A^{GM_5}(\eta)\right] - \Pr\left[\mathrm{Succ}_A^{GM_4}(\eta)\right] + \Pr\left[\mathrm{Succ}_A^{GM_4}(\eta)\right] - \Pr\left[\mathrm{Succ}_A^{GM_3}(\eta)\right] \\ &\quad + \Pr\left[\mathrm{Succ}_A^{GM_3}(\eta)\right] - \Pr\left[\mathrm{Succ}_A^{GM_2}(\eta)\right] + \Pr\left[\mathrm{Succ}_A^{GM_2}(\eta)\right] - \Pr\left[\mathrm{Succ}_A^{GM_1}(\eta)\right] \\ &= \frac{(t_{\mathrm{send}} + t_{\mathrm{exe}})^2}{2^u} + \mathrm{Adv}_A^K(\eta) + t_{\mathrm{hash}}^2 2^l + \max\left\{D' \cdot \frac{t_{\mathrm{send}}^{X'}, t_{\mathrm{send}}}{2^l}\right\}.\end{aligned} \tag{27}$$

So, we come to the final conclusion $\mathrm{Adv}_A^P(\eta) \le (t_{sen\ d} + t_{exe})^2\ /2^{u-1} + 2\,\mathrm{Adv}_A^K(\eta) + t_{\mathrm{hash}}^2\ \cdot 2^{l-1}\ + 2\max\{D' \cdot t_{\mathrm{send}}^{X}\ \prime, t_{\mathrm{send}}\ /2^l\}$. □

5.3. Informal Security Analysis. In this section, we further show that the proposed scheme is secure against the following attacks.

5.3.1. Privileged-Insider Attack. In this protocol, even if the attacker obtains the information $\{DI\ D_c, BRPW_c\}$ of the user in the registration process and the information $\{\beta_c, \gamma_c, \chi_c, \eta_c\}$ in the smart card, they cannot successfully obtain the session key. Because $SK = h(ID_c \oplus \alpha_c \oplus y_c \oplus T_1 \oplus T_2)$ and the user's ID_c is encrypted by ds before being transmitted to the server, even if the attacker obtains the value of $DI\ D_c$ and $BRPW_c$, the attack is futile. Therefore, this protocol can resist privileged-internal attacks.

5.3.2. Offline Password-Guessing Attacks. Suppose the attacker gets the message in the smart card; then, based on this message, they can guess the password offline. Even if the η_c value in the smart card is obtained and the values of ID_c and PW_c are guessed, the offline password-guessing operation cannot be successful. This is because the calculation of m also involves the value of the user's biological information R_i, and the value of R_i is difficult to obtain. Therefore, this protocol can effectively resist offline password-guessing attacks.

5.3.3. Replay Attack. Suppose that the malicious attacker intercepts the login information $\{DI\ D_c, \omega_c, \nu_c, T_1\}$ and authentication information $\{\mu_c, T_2\}$ and attempts to replay the login request. The request is invalid because we use the timestamp T_1 in the protocol to verify whether the time difference is within the set time threshold. Similarly, if the attacker intercepts the authentication message and attempts to make the authentication request, the user will also test the validity of the timestamp. Therefore, the protocol can effectively resist replay attacks.

5.3.4. Forward Secrecy. Assuming that the attacker obtains the value of the long-term password ds, they can only use this value to decrypt $DI\ D_c$ to obtain the value of the user's ID_c. However, because $SK = h(ID_c \oplus \alpha_c \oplus y_c \oplus T_1 \oplus T_2)$, it is not

Table 2: Comparisons of security.

Protocols	A1	A2	A3	A4	A5	A6
Rana et al. [19]	No	No	Yes	Yes	Yes	Yes
Kaul and Awasthi [18]	Yes	Yes	Yes	Yes	Yes	No
Xue et al. [29]	Yes	No	Yes	Yes	Yes	Yes
Lin et al. [30]	Yes	Yes	Yes	Yes	Yes	No
Chang et al. [17]	No	No	No	Yes	Yes	No
Kumari et al. [16]	No	Yes	No	Yes	Yes	Yes
Ours	Yes	Yes	Yes	Yes	Yes	Yes

sufficient to only know the value of the user's ID_c. The values of the parameters α_c and y_c cannot be obtained. Therefore, this protocol can provide perfect forward security.

5.3.5. Known Session-Specific Temporary Information Attacks. Assuming that the attacker obtains the value of temporary session information m or y_c, the session key cannot be obtained successfully. Because the session key calculation is composed of ID_c, but ID_c is encrypted by long-term key ds, the ID_c cannot be obtained by the attacker. Therefore, this protocol can successfully resist known session-specific temporary information attacks.

5.3.6. User Impersonation Attacks. Suppose that the attacker wants to carry out a user impersonation attack. They must first obtain the value of ID_c, but ID_c is encrypted by the long-term key ds, and so, it is difficult for the attacker to obtain its value. In addition, assuming that the attacker intercepts the message $\{DI\ D_c, \omega_c, \nu_c, T_1\}$ from the public channel and sends it to the server for verification, the user needs a certain amount of time to decrypt $DI\ D_c$. Therefore, when the server receives the message from the attacker for verification of the timestamp, it will find that the timestamp exceeds the set time domain and reject the login request. In this way, our protocol successfully resists user impersonation attacks.

5.3.7. Mutual Authentication. In this protocol, users and servers can successfully authenticate each other. First of all, the server authenticates the user through the value of v_c sent by the user. Similarly, the user can verify whether the server is legitimate through the value of μ_c sent by the server. Only legitimate users and servers can pass the authentication. Therefore, this protocol can effectively provide mutual authentication between users and servers.

6. Security and Performance Comparisons

This section discusses the security and performance analysis of the proposed protocol. Security analysis is mainly conducted through a comparison with other proposed protocols in the resistance of some common attacks, and performance analysis is mainly performed through a comparison with the time and communication costs of other protocols.

6.1. Security Comparisons. In this section, the protocol proposed in this study is compared with recent related protocols. Owing to the development of different types of attack technology and methods, previous protocols are now incapable of resisting some common attacks. At present, the common network attacks include A1: privileged-internal attack, A2: offline password-guessing attack, A3: replay attack, A4: perfect forward secrecy, A5: known session-specific temporary information attacks, and A6: user impersonation attacks. The comparison results are presented in Table 2. A "Yes" means that the protocol can resist the attack, whereas a "No" means that it cannot.

While the other related protocols each fail in some of the security attacks mentioned above, our proposed protocol can resist all the attacks, making our proposed protocol more secure and reliable.

6.2. Performance Comparisons. To better analyze the performance of this protocol, we compared it with a previous protocol. To obtain more convincing results, we analyzed the protocol using the same tools and under the same conditions and used the data provided by Rana et al. [19] The results show that different protocols have different execution times in the same execution environment. The time required for the connection operation and the noncollision hash function was 0.00014 ms and 0.00089 ms, respectively. The time required for the exception and encryption and decryption operations was extremely small, and so, it was not calculated. In addition, the number of bits required for the user name, password, arbitrary number, and integer was 160; the number of bits required for the private key and public key of the server was 256; the number of bits required for encryption and decryption was 512; and, the number of bits required for the exclusive or operation and noncollision hash function was 160 and 256, respectively. The symbols for each encryption operation are as follows:

$T_{\|}$: time required for connection operation

$T_{\oplus}$: time required for XOR operation

$T_{\text{Enc/Dec}}$: time required for encryption/decryption

T_h: time required for hash operation

First, we compared the communication cost of our proposed protocol with that of previous protocols. In particular, our protocol was compared with those proposed by Rana et al. [19], Kaul and Awasthi [18], Khan et al. [31], Chang et al. [17], and Kumari et al. [16]. The communication overhead of our protocol is 3136 bits, whereas that of the protocols proposed by Rana et al. [19], Kaul and Awasthi [18], Khan et al. [31], Chang et al. [17], and Kumari et al. [16] are 3296, 2668, 3744, 2336, and 3296 bits,

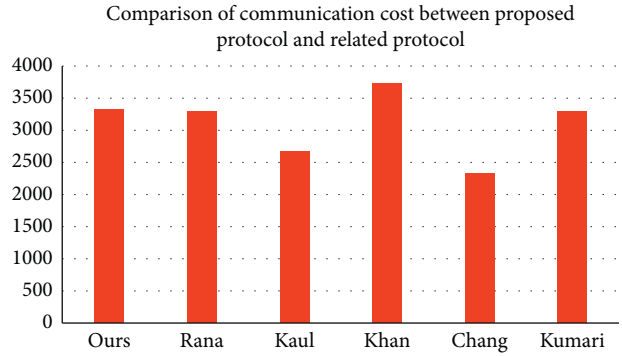

Figure 4: Communication cost.

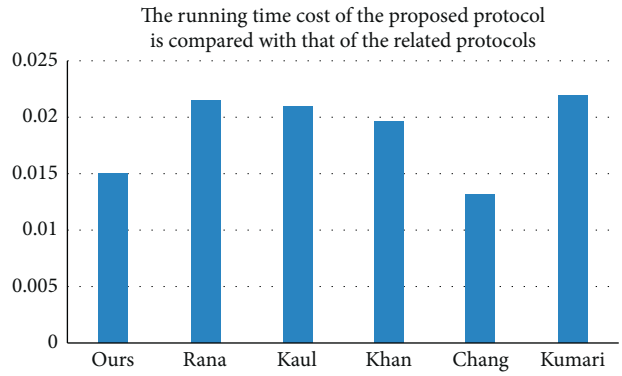

Figure 5: Running time.

Table 3: Proposed protocol comparison with related protocols.

	Running cost (ms)	Communication cost (bits)
Ours	$14T_h + 31T_\oplus + 19T_\parallel + 1T_{(\text{Enc/Dec})}$	3136
Rana et al. [19]	$20T_h + 29T_\oplus + 27T_\parallel + 3T_{(\text{Enc/Dec})}$	3296
Kaul and Awasthi [18]	$20T_h + 28T_\oplus + 23T_\parallel$	2668
Khan et al. [31]	$15T_h + 11T_\oplus + 45T_\parallel + 4T_{(\text{Enc/Dec})}$	3744
Chang et al. [17]	$12T_h + 7T_\oplus + 18T_\parallel$	2336
Kumari et al. [16]	$19T_h + 18T_\oplus + 36T_\parallel$	3296

respectively. As shown in Figure 4, the communication cost of our protocol is lower than that of Rana et al. and Khan et al., but slightly higher than that of Kaul and Awasthi [18]. Although the communication cost of Chang et al. is small, the protocol proposed by them cannot effectively resist privilege internal attacks, offline password guessing attacks, and replay attacks.

Next, we compare the running time cost of our proposed protocol with those of the three protocols mentioned above. The operating cost of our protocol is 0.01512 ms, whereas that of the protocols proposed by Rana et al. [19], Kaul and Awasthi [18], Khan et al. [31], Chang et al. [17], and Kumari et al. [16] are 0.0215 ms, 0.021 ms, 0.01965 ms, 0.01318 ms, and 0.02191 ms, respectively. As shown in Figure 5, the running time of our proposed protocol is shorter than that of the four protocols mentioned above. Although the time consumption of the protocol proposed by us is a little higher than that proposed by Chang et al., the protocol proposed by Chang et al. has the problem of security. It can be said that our protocol has better performance than the ones mentioned above.

Through the analysis of Tables 2 and 3, our protocol is slightly higher than Kaul and Awasthi's [18] protocol in terms of communication cost, but Kaul and Awasthi's [18] protocol cannot resist user simulation attacks. Because our proposed protocol can more effectively resist various security attacks, our protocol is more applicable in future works.

7. Conclusions

In this study, we analyzed the next generation Internet of Things remote protocol proposed by Rana et al., and found that their protocol cannot resist all kinds of security attacks as they claim. Specifically, we found that their protocols are vulnerable to offline password-guessing attacks and

privileged-insider attacks. To solve these problems, we introduced a three-factor security protocol utilizing biological information. In addition, we proved the security and reliability of the protocol through BAN logic and ROR analysis. Finally, we compared the proposed protocol with the previous related protocols and found that our protocol is better in terms of both communication cost and time cost. Therefore, our proposed protocol is more applicable and referential for the development of the future work.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] E. K. Wang, C.-M. Chen, M. M. Hassan, and A. Almogren, "A deep learning based medical image segmentation technique in internet-of-medical-things domain," *Future Generation Computer Systems*, vol. 108, pp. 135–144, 2020.

[2] C.-T. Li, C.-C. Lee, C.-Y. Weng, and C.-M. Chen, "Towards secure authenticating of cache in the reader for RFID-based IoT systems," *Peer-to-Peer Networking and Applications*, vol. 11, no. 1, pp. 198–208, 2018.

[3] H. Xiong, Y. Zhao, Y. Hou et al., "Heterogeneous signcryption with equality test for IIoT environment," *IEEE Internet of Things Journal*, 2020.

[4] T. Y. Wu, T. Wang, Y. Q. Lee, W. Zheng, S. Kumari, and S. Kumar, "Improved authenticated key agreement scheme for fog-driven IoT healthcare system," *Security and Communication Networks*, vol. 2021, Article ID 6658041, 16 pages, 2021.

[5] T.-Y. Wu, Z. Lee, L. Yang, C.-M. Chen, and R. Tso, "A provably secure authentication and key exchange protocol in vehicular ad hoc networks," *Security and Communication Networks*, vol. 2021, Article ID 9944460, 17 pages, 2021.

[6] P. Wang, C. M. Chen, S. Kumari, M. Shojafar, R. Tafazolli, and Y. N. Liu, "HDMA: hybrid D2D message authentication scheme for 5G-enabled VANETs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, 2020.

[7] Z. Ali, A. Ghani, I. Khan, S. A. Chaudhry, S. H. Islam, and D. Giri, "A robust authentication and access control protocol for securing wireless healthcare sensor networks," *Journal of Information Security and Applications*, vol. 52, Article ID 102502, 2020.

[8] Q. Jiang, J. Ma, C. Yang, X. Ma, J. Shen, and S. A. Chaudhry, "Efficient end-to-end authentication protocol for wearable health monitoring systems," *Computers & Electrical Engineering*, vol. 63, pp. 182–195, 2017.

[9] M. F. Ayub, K. Mahmood, S. Kumari, and A. K. Sangaiah, "Lightweight authentication protocol for e-health clouds in IoT-based applications through 5G technology," *Digital Communications and Networks*, vol. 7, no. 2, pp. 235–244, 2021.

[10] A. T. Khan, S. Li, and X. Cao, "Control framework for cooperative robots in smart home using bio-inspired neural network," *Measurement*, vol. 167, Article ID 108253, 2021.

[11] X. Chen, A. Li, X. e. Zeng, W. Guo, and G. Huang, "Runtime model based approach to IoT application development," *Frontiers of Computer Science*, vol. 9, no. 4, pp. 540–553, 2015.

[12] C. M. Chen, L. Chen, Y. Huang, S. Kumar, and J. M. T. Wu, "Lightweight authentication protocol in edge-based smart grid environment," *EURASIP Journal on Wireless Communications and Networking*, vol. 2021, no. 1, pp. 1–18, 2021.

[13] R. Vinoth, L. J. Deborah, P. Vijayakumar, and N. Kumar, "Secure multifactor Authenticated key agreement scheme for industrial IoT," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3801–3811, 2020.

[14] Y. Yu, L. Hu, and J. Chu, "A secure authentication and key agreement scheme for IoT-based cloud computing environment," *Symmetry*, vol. 12, no. 1, p. 150, 2020.

[15] M. Soni and D. K. Singh, "LAKA: lightweight Authentication and key agreement protocol for Internet of things based wireless body area network," *Wireless Personal Communications*, vol. 1–18, 2021.

[16] S. Kumari, M. K. Khan, and X. Li, "An improved remote user authentication scheme with key agreement," *Computers & Electrical Engineering*, vol. 40, no. 6, pp. 1997–2012, 2014.

[17] Y. F. Chang, W. L. Tai, and H. C. Chang, "Untraceable dynamic-identity-based remote user authentication scheme with verifiable password update," *International Journal of Communication Systems*, vol. 27, no. 11, pp. 3430–3440, 2014.

[18] S. D. Kaul and A. K. Awasthi, "Security enhancement of an improved remote user authentication scheme with key agreement," *Wireless Personal Communications*, vol. 89, no. 2, pp. 621–637, 2016.

[19] M. Rana, A. Shafiq, I. Altaf et al., "A secure and lightweight authentication scheme for next generation IoT infrastructure," *Computer Communications*, vol. 165, pp. 85–96, 2021.

[20] M. Burrows, M. Abadi, and R. M. Needham, "A logic of authentication," *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, vol. 426, no. 1871, pp. 233–271, 1989.

[21] D. Wang, D. He, P. Wang, and C. H. Chu, "Anonymous two-factor authentication in distributed systems: certain goals are beyond attainment," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 4, pp. 428–442, 2014.

[22] D. Wang and P. Wang, "Two birds with one stone: two-factor authentication with security beyond conventional bound," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 708–722, 2016.

[23] P. Kocher, J. Jaffe, B. Jun, and P. Rohatgi, "Introduction to differential power analysis," *Journal of Cryptographic Engineering*, vol. 1, no. 1, pp. 5–27, 2011.

[24] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Transactions on Computers*, vol. 51, no. 5, pp. 541–552, 2002.

[25] D. Kumar, H. S. Grover, D. Kaur, A. Verma, K. K. Saini, and B. Kumar, "An efficient anonymous user authentication and key agreement protocol for wireless sensor networks," *International Journal of Communication Systems*, vol. 34, no. 5, Article ID e4724, 2021.

[26] S. Kumari and K. Renuka, "Design of a password authentication and key agreement scheme to access e-healthcare services," *Wireless Personal Communications*, vol. 117, pp. 1–19, 2019.

[27] D. Kang, H. Lee, Y. Lee, and D. Won, "Lightweight user authentication scheme for roaming service in GLOMONET

with privacy preserving," *PLoS One*, vol. 16, no. 2, Article ID e0247441, 2021.

[28] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipf's law in passwords," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2776–2791, 2017.

[29] K. Xue, P. Hong, and C. Ma, "A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture," *Journal of Computer and System Sciences*, vol. 80, no. 1, pp. 195–206, 2014.

[30] I.-C. Lin, M.-S. Hwang, and L.-H. Li, "A new remote user authentication scheme for multi-server architecture," *Future Generation Computer Systems*, vol. 19, no. 1, pp. 13–22, 2003.

[31] A. A. Khan, V. Kumar, M. Ahmad, and S. Rana, "LAKAF: lightweight authentication and key agreement framework for smart grid network," *Journal of Systems Architecture*, vol. 116, Article ID 102053, 2021.